Microlog Microfiche Collection
Resolution Chart

Copyright

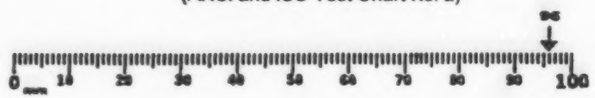
Copyright for the ***contents*** of this microfiche is held by the publisher of the original document. Please contact the original publisher for guidance in all copyright matters. Copyright in this ***microfiche edition*** is held by Micromedia Limited.

Quality of the Microfiche Reproduction

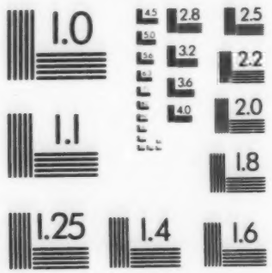
Micromedia Limited maintains the highest standards for micrographic reproduction. Attempts are made to obtain the best quality print version to microfilm. However, there are some printing styles, colours and graphic layout styles (including maps, graphs and tables) which do not reproduce well. Because of the value of the original document, Micromedia has filmed it recognizing that it may not be as easy to read as the original document.

Some pages have been filmed twice to improve the clarity of the reproduction. Consult both reproductions to ensure that all of the original text is read.

Microcopy Resolution Test Chart
National Bureau of Standards
Standard Reference Material 1010a
(ANSI and ISO Test Chart No. 2)



PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT



MS109

24x

micromedia
a division of IHS Canada

20 Victoria Street, Toronto, ON M5C 2N8
Tel.: (416) 362-5211
Toll free: 1-800-387-2689
Fax: (416) 362-6161

Collection MICROLOG sur Micromédia
Tableau de résolution

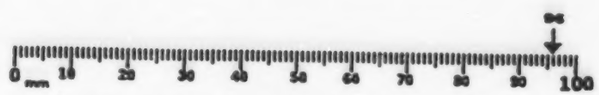
Droits d'auteurs

Les droits d'auteurs du conenu de cette microfiche sont détenus par l'éditeur du document original. Veuillez communiquer avec celui-ci pour obtenir des renseignements sur tout ce qui touche les droits d'auteurs. Les droits d'auteurs de cette édition sur microfiche sont détenus par Micromédia Limitée.

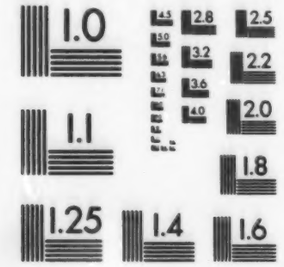
Qualité de reproduction sur microfiche

Micromédia Limitée maintient les normes de reproduction micrographique les plus élevées. Elle fait tout son possible pour obtenir la version sur papier à reproduire sur microfiche de la meilleure qualité possible. Cependant, certains styles d'imprimerie, de couleurs et de style de mise en page graphique (incluant des cartes, des grahiques et des tableaux) se reproduisent mal. En raison de la valeur du document original, Micromédia l'a reproduit tout en reconnaissant qu'il ne sera peut-être pas aussi lisable que le document original.

Tableau de résolution micrographique
National Bureau of Standards
Standard Reference Material 1010a
(ANSI and ISO Test Chart No. 2)



PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT



MS109

24x

micromedia
a division of IHS Canada

20 Victoria Street, Toronto, ON M5C 2N8
Tel.: (416) 362-5211
Toll free: 1-800-387-2689
Fax: (416) 362-6161

CRT-2001-32

Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

PARALLEL COMPUTATION, CO-OPERATION, TABU SEARCH

by

Teodor Gabriel Crainic

September 2001

102-02947/

PUBLICATION
CRT-2001-32

PARALLEL COMPUTATION, CO-OPERATION, TABU SEARCH*

by

Teodor Gabriel Crainic[1]

* This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR).

[1] Département de management et technologie, Université du Québec à Montréal, et Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2001

ABSTRACT

We present strategies to design parallel tabu search algorithms and survey developments and results in the area. In the second part of the paper, we focus on multisearch strategies. We discuss general design and implementation principles, point out a number of challenges and pitfalls, and identify trends and promising research directions.

Key words : parallel computation, parallelization strategies, co-operative search, tabu search, scatter search, path relinking.

RÉSUMÉ

Nous présentons et discutons des principes généraux de definition et de mise en oeuvre des recherches avec tabous parallèles, ainsi qu'une revue de l'état de l'art dans le domaine. Dans la deuxième partie de l'article, nous regardons de façon particulière les stratégies multi-processus. Nous identifions des défis et des pièges, des tendances majeures, ainsi que des directions de recherche offrant des perspectives intéressantes.

Mots-clés : calcul parallèle, stratégies de parallélisation, méthodes de recherche coopératives, méthodes de recherche avec tabous, méthodes de "scatter search" et de "path relinking".

1 Introduction

Parallel computation is increasingly acknowledged as a natural complement to problem solving strategies, particularly when applied to large-scale, hard formulations and instances. The field of meta-heuristics is no exception to this trend and the last ten to fifteen years have witnessed an increasingly strong stream of important developments.

Tabu search has offered, and continues to offer, a particularly rich field for the development and experimentation of parallel meta-heuristics. Simultaneously, major efforts aimed other meta-heuristic classes, primarily genetic, "natural" evolution-inspired algorithms and simulated annealing methods. A number of these efforts targeted particular problem instances and yielded tailor-made procedures. The goal of most developments is, however, to propose strategies with a larger potential utilization, at least within the framework of the underlying (sequential) search methodology. These strategies are the object of this paper.

The main goals of parallel meta-heuristics join those of most efforts to develop parallel computing methods: To solve the problem quicker than the sequential method does. In meta-heuristic terms, these goals may be stated as: Compared to the sequential method to 1) accelerate the search for a comparable solution quality; and 2) broaden the search, that is, to obtain better solutions for a comparable computation effort or, at least, "wall-clock" time. Because meta-heuristics are un-exact resolution methods, a third goal is increasingly assigned to parallel strategies. Parallel meta-heuristic search methods should be more robust than sequential methods with respect to the characteristics of the problem instances they encounter. It is thus expected not only that parallel implementations identify better solutions but also that they do so consistently over diverse sets of problem instances. And that this performance is ensured without excessive calibration efforts.

On examining the literature, one realizes that beyond the idiosyncrasies of each implementation that follow from the particular problem instance addressed, there are only a limited number of general parallel strategies that may be used to build parallel meta-heuristics. A number of these strategies are well-understood by now. Some are still a fertile ground for intense development and offer exiting perspectives on our ability to address hard and large-scale problem formulations. In this paper we focus on tabu search and scatter search. The objectives are to briefly review parallel strategies and developments for tabu search, to discuss in somewhat more details cooperative methods that are the subject of most contemporary efforts, and to identify promising research directions.

The paper is organized as follows. Section 2 describes parallel strategies and re-

views parallel tabu search methods proposed in the literature. Section 3 describes the main multi-search strategies and discusses the challenges associated to the development of successful co-operative methods. It also presents a number of ideas on how parallel strategies could be applied to some advanced tabu search methods, particularly path relinking and scatter search. Section 4 sums up the paper with a number of interesting challenges and research directions for parallel tabu search.

2 Parallel Tabu Search

Meta-heuristics have been defined as master strategies (heuristics) to guide and modify other heuristics to avoid getting trapped in local optima or sequences of visited solutions (*cycling*), produce solutions beyond those normally identified by local search heuristics, and provide reasonable assurance that the search has not overlooked promising regions (Glover 1986; see also Glover and Laguna 1993).

Tabu search, similarly to most meta-heuristics, is an *improving* iterative procedure that *moves* from a given solution to a solution in its *neighbourhood* that is better in terms of the objective function value (or some other measure based on the solution characteristics). Thus, at each iteration, a *local search* procedure (the "original" heuristic) identifies and evaluates solutions in the neighbourhood of the current solution, selects the best one relative to given criteria, and implements the transformations required to establish the selected solution as the current one. Inferior quality solutions encountered during neighbourhood explorations may be accepted as a strategy to move away from local optima. In theory, the procedure iterates until no further improvement is possible. In actual implementations, a more restrictive stopping criteria is used: total number of iterations, relative improvement over a number of iterations, etc.

Memory and *memory hierarchy* are major concepts in tabu search, as are the memory-based strategies used to guide the procedure into various search phases, possibly using different neighbourhoods. Short-term *tabu status* memories record recent visited solutions or their attributes to avoid search cycling due to the repetition or inversion of recent actions. The tabu status of a move may be lifted if testing it against an *aspiration criterion* signals the discovery of a high quality solution (typically, the best one encountered so far). Medium to long-term memory structures record various informations and statistics relative to the solutions already encountered (e.g., frequency of certain attributes in the best solutions) to "learn" about the solution space and guide the search. *Intensification* of the search around good solutions and its *diversification* towards regions of the solution space not yet explored are two main ingredients in tabu search. These two types of moves are based on medium

and long-term memories, are implemented using specific neighbourhoods, and may involve quite complex solution transformations. More details on the basic and advanced features of tabu search may be found in Glover (1986, 1989, 1990, 1996) and Glover and Laguna (1993, 1997).

Tabu search has proved a fertile ground for innovation and experimentation in the area of parallel meta-heuristics. Most parallel strategies encountered in the literature have been applied to tabu search for a variety of applications and a number of interesting parallelization concepts have been introduced while developing parallel tabu search methods (Crainic and Toulouse 2001).

Crainic, Toulouse, and Gendreau (1997) introduced in 1993 what is still the most comprehensive taxonomy of parallel tabu search methods. The classification has three dimensions (Figure 1). The first dimension, *Search Control Cardinality*, explicitly examines how the global search is controlled, by a single process (as in master-slave implementations) or collegially by several processes that may collaborate or not. The two alternatives are identified as *1-control (1C)* and *p-control (pC)*, respectively. The classes of the second dimension indicate the *Search Differentiation*: do search threads start from the same or different solutions and do they make use of the same or different search strategies? The four cases considered are: *SPSS, Same initial Point, Same search Strategy*; *SPDS, Same initial Point, Different search Strategies*; *MPSS, Multiple initial Points, Same search Strategies*; *MPDS, Multiple initial Points, Different search Strategies* (see also Voß 1993). Finally, the dimension relative to the type of *Search Control and Communications* addresses the issue of how information is exchanged.

In parallel computing, one generally refers to *synchronous* and *asynchronous* communications. In the former case, all processes have to stop and engage in some form of communication and information exchange at moments (number of iterations, time intervals, specified algorithmic stages, etc.) exogenously determined, either hard-coded into the procedures or determined by a control (master) process. In the latter case, each process is in charge of its own search, as well as of establishing communications with the other processes, and the global search terminates once each individual search stops. To reflect more adequately the quantity and quality of information exchanged and shared, as well as the additional knowledge derived from these exchanges (if any), we refine these notions and divide methods into four classes. *Rigid (RS)* and *Knowledge Synchronization (KS)* and, symmetrically, *Collegial (C)* and *Knowledge Collegial (KC)*. Crainic, Toulouse, and Gendreau (1997) detail the taxonomy and use it to analyze the parallel methods proposed in the literature up to that time.

Typically, 1-control strategies implement a classical master-slave approach aimed solely towards the acceleration of the search. Here, a "master" processor executes a

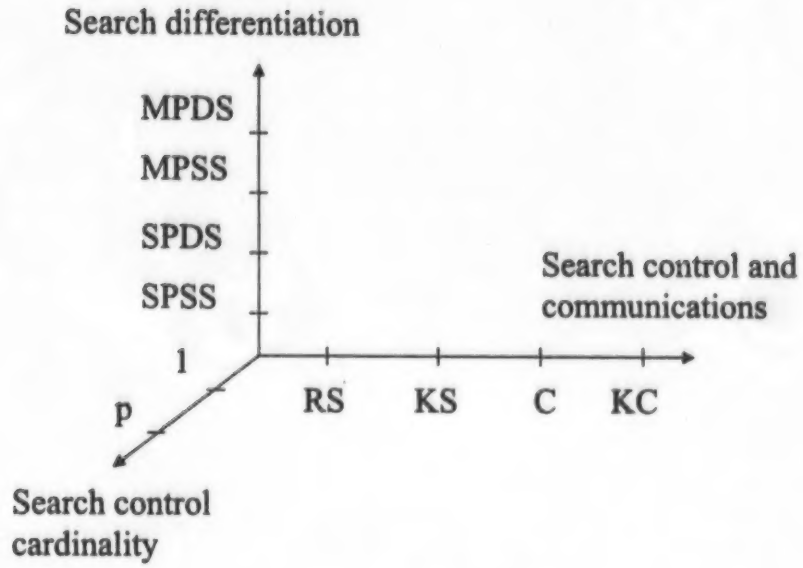


Figure 1: Method-oriented Taxonomy

synchronous tabu search procedure but dispatches computing-intensive tasks to be executed in parallel by "slave" processes. The master receives and processes the information resulting from the slave operations, selects and implements moves, gathers all the information generated during the tabu exploration, updates the memories, and decides whether to activate different search strategies or stop the search.

In the context of tabu search, the operation most widely targeted in such approaches is the neighbourhood evaluation: At each iteration, the possible moves in the neighbourhood of the current solution are partitioned into as many sets as the number of available processors and the evaluation is carried on in parallel by slave processes. This 1C/RS/SPSS strategy yielded very interesting results for formulations with large neighbourhoods and relatively small computing efforts required to evaluate and perform a given move is, as in *quadratic assignment* (*QAP*: Chakrapani and Skorin-Kapov 1992, 1993, 1995; Taillard 1991, 1993a), *travelling salesman* (*TSP*: Chakrapani and Skorin-Kapov 1993a) and *vehicle routing* (*VRP*: Garcia, Potvin, and Rousseau 1994) applications. For the same quality of solution, near-linear speedups are reported using a relatively small number of processors. Moreover, historically, this approach permitted improvements to the best-known solutions to several problem instances proposed in the literature.

Implementations of the *sequential fan candidate list* strategy (Glover, Taillard, de Werra 1993; Glover and Laguna 1997), also known as *look ahead* or *probing* approaches, allow slave processes to perform a number of iterations before synchronization and the selection of the best neighbouring solution from which the next iteration is initiated. Probing strategies thus belong to the 1-control, knowledge synchronous (1C/KS) class and may use any of the four search differentiation models identified earlier on. The only parallel implementation of this strategy the author is aware of is to be found in the comparative study of several synchronous tabu search parallelizations performed by Crainic, Toulouse, and Gendreau (1995a) for the location-allocation problem with balancing requirements. In this study, the authors implemented and compared a 1C/RS/SPSS and a 1C/KS/SPSS method. The second performed marginally better. However, both methods were outperformed by p-control implementations that attempt a more thorough exploration of the solution space. These results also emphasize that performance is less interesting for the low-level parallelism offered by single control, synchronous parallel strategies when the time required by one serial iteration is relatively important compared to the total solution time and only a few hundred moves may be executed in a reasonable computing time compared to the tens of thousands performed by a typical VRP tabu search procedure.

Domain decomposition is another major parallelization strategy. The fundamental idea is simple: Partition the feasible domain of the problem in several (disjoint) sets

and execute the contemplated meta-heuristic on each subset, thus accelerating the global search. Two cases may then be encountered. Either the solution of each search thread is a complete solution to the problem in hand or a partial solution only, in which case a complete solution has to be reconstructed. In both cases, however, each search process has had access to a part of the domain only (especially when the partition is strictly enforced during the search). Then, to somewhat increase the thoroughness of the search, the partition is modified and the search is started anew. And the process is repeated a number of times.

The "natural" implementation of domain decomposition is through a 1C/KS master-slave scheme with any of the search differentiation strategy. (Notice that different meta-heuristics may be applied to different sets and that, by definition, each search starts from a different initial point.) The master determines the partition, synchronizes slave processes, reconstructs solutions (when required), and determines stopping conditions. Slave processes perform the search on their assigned partitions. Such approaches have been successful for problems for which numerous iterations may be performed in a relatively short time and restarting the method with several different partitions does not require unreasonable computational efforts (e.g., Fiechter 1994 for the *TSP*; Porto and Ribeiro 1995, 1996 and Porto, Kitajima, and Ribeiro 2000 for the task scheduling problem on heterogeneous systems)

The same idea may also be implemented in a pC/KS framework, however. In this case, processors stop at pre-determined moments to exchange partial solutions and build whole ones, modify the partition, verify the stopping conditions. From a computer programming point of view, one of the processors may assume co-ordination tasks, but this does not change the particular nature of the algorithmic design. Taillard's (1993) early tabu search for VRP follows this pC/KS/MPSS paradigm. The domain is partitioned and vehicles are allocated to the resulting regions. Once the initial partition is performed, each subproblem is solved by an independent tabu search. All processors stop after a number of iterations that varies according to the total number of iterations already performed. The partition is then modified by an information exchange phase, during which tours, undelivered cities, and empty vehicles are exchanged between adjacent processors (corresponding to neighbouring regions).

Partition approaches did allow to address successfully a number of problem instances. The synchronization inherent in the design of the strategy hinder its performance, however. Indeed, in most cases, results obtained by using partition strategies have been improved by methods that launch several search threads to simultaneously explore the solution space with various degrees of co-operation. In fact, enjoying the benefit of hindsight, the main contribution of the Taillard (1993) paper is to mark the evolution towards one of the most successful sequential meta-heuristics for the VRP, the *adaptive memory* tabu search (Rochat and Taillard 1995; Glover 1996).

According to an adaptive memory approach applied to the VRP, cities are initially separated into several subsets, and routes are built using a construction heuristic. Initial routes are then stored in a structure called an *adaptive memory*. Then, a combination procedure builds a complete solution using the routes in the memory, and the solution is further improved using a tabu search method. The routes of "good" solutions are then deposited into the same memory, which thus adapts to reflect the current state of knowledge of the search. The process then re-starts with a new solution built from the routes stored in the adaptive memory. The method stops when a pre-specified number of calls to the adaptive memory have been performed. This approach clearly implements the principles of decomposition using a serial procedure and has now been successfully applied to other problem classes. However, interestingly, most parallel applications of this approach are now found in co-operative multi-thread strategies.

Multi-thread parallelizations for tabu search methods follow the same basic pattern: p threads search through the same solution space, starting from possibly different initial solutions and using possibly different tabu (or other) search strategies. Historically, independent and synchronous co-operative multi-thread methods were proposed first. Currently, asynchronous procedures are being increasingly developed and one observes an increased awareness of the issues related to the definition and modelling of co-operation (see Section 3).

Independent multi-thread searches belong to the pC/RS class of the taxonomy. Most implementations start several independent search processes from different, randomly generated, initial configurations. No attempt is made to take advantage of the multiple threads running in parallel other than to identify the best overall solution being identified once all processes stop. This definitively earns independent search strategies their Rigid Synchronization classification. (Note that, in general, the implementations designate a processor to collect the information and verify stopping criteria.) Battiti and Tecchiolli (1992, for the QAP), Taillard (the main study is found in his 1994 paper on parallel tabu methods for job shop scheduling problems) and others studied and empirically established the efficiency of such procedures when compared to the best heuristics proposed at the time for their respective problems. This parallelization of the classic sequential multi-start heuristic is easy to implement and may offer satisfactory results. Co-operative strategies often offer superior performance, however (e.g., Crainic, Toulouse, and Gendreau 1995b, Crainic and Gendreau 2001).

pC/KS strategies are also generally implemented in a master-slave setting but attempt to take advantage of the parallel explorations by synchronizing processors at pre-determined intervals. The master process than collects information and usually restarts the search from the best solution (Malek *et al.* 1989, for the TSP; Rego and

Roucairol 1996 for the VRP based using ejection chains). De Falco *et al.* (1994, for the QAP), and De Falco, Del Balio, and Tarantino (1995, for the mapping problem) attempted to overcome the limitations of the master-slave setting. When each search thread terminates its local search, they synchronize and best solutions are exchanged between processes that run on neighbouring processos. Good performance and results were reported by all authors.

Asynchronous co-operative multi-thread search methods belong to the pC/C or pC/KC classes of the taxonomy according to the quantity and quality of the information exchanged and, eventually, on the "new" knowledge inferred based on these exchanges. Most such developments we identified use some form of *memory* (the term *blackboard* is also used sometimes) for inter-thread communications. Each individual search thread starts from (usually) a different initial solution and generally follows a different search strategy. Exchanges are performed asynchronously and through the memory. Memories recording the performance of individual solutions, solution components, or even search threads may be added to the pool and statics may be gradually built. Moreover, various procedures may also be added to the pool to attempt to extract information or to create new informations and solutions based on the solutions exchanged. Co-operative multi-thread strategies implementing such methods belong to the pC/KC class.

One may classify co-operative multi-thread search methods according to the type of information stored in the central memory: complete or partial solutions. In the latter case, one often refers to *adaptive memory* strategies, while *central memory*, *pool of solutions*, *solution warehouse* or, even, *reference set* are terms used for the former. These methods have proved extremely successful on a broad range of problem types (Gendreau *et al.* 1999 for real-time routing and vehicle dispatching; Taillard *et al.* 1997, Badeau *et al.* 1997, Schulze and Fahle 1999, Le Bouthiller and Crainic 2001 for VRP with time windows; Crainic, Toulouse, and Gendreau 1995 for multicommodity location with balancing requirements; Crainic and Gendreau 2001 for the fixed cost, capacitated, multicommodity network design problem; Andreatta and Ribeiro 1994, Aiex *et al.* 1996, and Martins, Ribeiro, and Rodriguez 1996 for the problem of partitioning integrated circuits for logical testing.) and we return to them in the next section.

One must notice, however, that the selection of the information exchanged, and the part of it that is propagated past the initial exchange, significantly impacts the performance of co-operating procedures. The determination for each search thread of both when to import external information and how to use it (e.g., restart from the imported solution and clean up all history contrasted to the use of imported global information to bias local selection mechanisms) are equally important ingredients in the design of co-operating multi-thread parallel schemes. The application of these

principles forms the basis of a new co-operation scheme, called *multi-level co-operative search* that has proven very successful for graph partitioning problems (Toulouse, Thulasiraman, and Glover 1999; see also Toulouse, Glover, and Thulasiraman 1998 and Ouyang *et al.* 2000, 2000a).

Although introduced for tabu search, the classification used in this paper applies to many other classes of meta-heuristics and refines the criteria based on the impact of parallelization on the search trajectory (e.g., Crainic and Toulouse 2001). Thus, it may form the basis for a comprehensive taxonomy of parallel meta-heuristics. This task, however, is beyond the scope of this paper.

3 Co-operation, Path Relinking, Scatter Search

A trend emerges from the collective efforts dedicated to parallel meta-heuristics in general and tabu search in particular. The main methodological focus moved from the low-level parallelism (e.g., the 1C/RS methods) of the initial developments, passing through a phase of domain decomposition approaches, on to independent search methods and co-operation (and hybridization). Of course, each type of parallelization strategy may play an important role given the particular problem class and instance. On the other hand, co-operation often offers superior performance but requires more careful design and implementation.

Co-operative multi-thread tabu search methods launch several independent searches and implement information exchanging mechanisms among these threads. The general objective is to exchange *meaningful* information in a *timely* manner such that the global, parallel search achieves a better performance than the simple concatenation of the results of the individual methods. Performance is measured in both computing time and solution quality.

As mentioned already, independent search methods implement multi-start sequential heuristics and, therefore, cannot achieve both performance objectives. Computation time is indeed reduced, but the same solution is reached. To overcame this limitation, Knowledge Synchronous pC strategies have been implemented where information is exchanged once all threads have completed their searches. The best overall solution is the only information exchanged and it serves as departure point for a new round of independent searches. The method is simple to implement and authors report good results. Computing times soar, however, and it would be interesting to thoroughly compare solution quality performance with co-operative methods. The limited evidence to this effect comes from applications to vehicle routing formulations and seem to indicate that co-operative strategies perform better in general.

Two other issues should be investigated in relation to these pC/KS strategies. First, the implementations reported in the literature start *new* searches following synchronization, cleaning up all memories. Given the importance of memories for tabu search, one wonders whether precious information is lost in the process. An investigation into what information to pass from one independent search phase to the next, how to use it, and its impact on performance might prove very interesting. The second issue has to do with the fact that, up to now, the only information exchanged when processes synchronize is the best solution. Yet, it has been shown that for co-operative methods this strategy may be counter-productive by concentrating most (all) threads on the same region of the solution space. This is an issue for further study as well.

Co-operation strategies require careful design as well as resources to manage the exchanges and process the data. A number of fundamental issues have to be addressed when designing co-operative parallel strategies for meta-heuristics (Toulouse, Crainic, and Gendreau 1996, Toulouse *et al.* 1998, Toulouse, Crainic, and Sansó 1999, Toulouse, Crainic, and Thulasiraman 2000):

- What information is exchanged?
- Between what processes it is exchanged?
- When is information exchanged?
- How it is exchanged?
- What does one do with imported data?

It is beyond the scope of this paper to address in detail all these issues. The reader may refer to the papers just mentioned for a more thorough description. It may be of interest, however, too sum up some of the pitfalls and less-than-winning strategies that have been identified in the literature so far.

Synchronous co-operative implementations tend to show a poorer performance compared to asynchronous and independent searches, especially when synchronization points are pre-determined (as in most current implementations). Such strategies display large computation overhead and loss of efficiency due to waiting for all processes to reach the synchronization point. The pre-defined synchronization points also makes them much less reactive to the environment and the progress of the search than asynchronous approaches and methods.

Asynchronous strategies are increasingly being proposed. Not all approaches offer the potential for success, however. To illustrate some of the issues, consider the

following generic "broadcast and replace" strategy. When a search thread improves its local best solution, a message is sent to all other processes to stop their own exploration and import the broadcast solution. Then, if the imported solution is better than the local one, the search is restarted with the new solution. Several variants of this approach may be found in the literature. It has been observed, however, that interrupting the "local search" phase does not yield good results. The main reason is that one no longer performs a thorough search based on the local observation of the solution space. Rather, one transforms the tabu search into random search. This phenomenon is particularly disruptive if one permits the initial local search phase to be interrupted.

Stated in a more general way, the observation is that co-operative meta-heuristics with unrestricted access to shared knowledge may experience serious premature "convergence" difficulties, especially when the shared knowledge reduces to one solution (the overall best or the new best from a given thread). This is due to a combination of factors: There is no global history or knowledge on the parallel search and thus each process has a (very) partial view of the entire search; Threads often use similar criteria to access the (same) "best" solution; Each process may broadcast a new best solution after a few moves only and thus disseminate information where only part of the solution has been properly evaluated. The contents of shared data tend then to become stable and biased by the best moves of the most performing threads and one observes a premature convergence of the dynamic search process. Moreover, the phenomenon may be observed whether one initializes the local memories following the import of an external solution or not.

Adaptive and *central* memory strategies attempt to overcame these limitations by implementing mechanisms to build knowledge regarding the global search that are then used to guide the exchange of information and, thus, to impact the trajectory of each individual process. The two approaches have much in common. In both approaches, there are no communications among the individual search processes: all communications take place between the memory process and individual search threads. Both approaches control when communications take place (not very often and not during local search phases) and assign this responsibility to the individual searches. Based on the implementations reported in the literature, differences are mainly at the level of the information stored in the memory and its use.

As indicated in the previous section, adaptive memory implementations follow the general principles of the sequential adaptive memory strategy (Rochat and Taillard 1995) and store the elements of the best solutions found by the individual threads together with, eventually, memories counting the frequency of each element in the best solutions encountered so far. Processes communicate their best solutions at the normal end of their search. New starting solutions are built out of the elements in

the memory. When frequency memories are implemented, they are used to bias the selection of the elements during the construction phase.

The central memory approach has less limitations. In theory at least since, in our opinion, the potential of this method has not been fully explored yet. As far as we can tell, Crainic, Toulouse, and Gendreau (1997) proposed the first central memory strategy for tabu search. A more fully developed implementation is reported by Crainic and Gendreau (2001) for the fixed cost, capacitated, multicommodity network design problem. In both studies, the individual tabu search threads communicate the vector of design variables each time the local solution is improved but import a solution from the central memory only before undertaking a diversification phase. Then, if the imported solution is better than the current best, the diversification proceeds from the new solution. Otherwise, the imported solution is discarded and the procedure proceeds as usual. Memories are never re-initialized. The authors compared five strategies of retrieving a solution from the pool when requested by an individual thread. The strategy that always returns the overall best solution displayed the best performance when few (4) processors were used. When the number of processors was increased, a probabilistic procedure, based on the rank of the solution in the pool, appears to offer the best performance. The parallel procedure improves the quality of the solution and also requires less (wall clock) computing time compared to the sequential version, particularly for large problems with many commodities.

Recently, Le Bouthiller and Crainic (2001) took this approach one step further and proposed a central memory parallel meta-heuristic for the VRP with time windows where several tabu search and genetic algorithm threads co-operate. In this model, the central memory constitutes the population common to all genetic threads. Each genetic algorithm has its own parent selection and crossover operators. The offspring are returned to the pool to be enhanced by a tabu search procedure. The tabu search threads and the central memory follow the same rules as in the work of Crainic and Gendreau (2001). Experimental results show that without any particular calibration, the parallel meta-heuristic obtains solutions whose quality is comparable to the best meta-heuristics available, and demonstrates almost linear speedups. These results indicate that central memory approaches apply equally well whether complex constraint structures complement the combinatorial characteristics of the problems studied or not.

Many enhancements could be added to these co-operation mechanisms and should be studied further. One such enhancement, proposed but not yet tested in the literature, concerns performance measures that could be attached to each individual search. Then, processes that do not contribute significantly neither to the quality of the solutions in the pool, nor to their diversity could be discarded. The computing resources thus recuperated could either be simply released or assigned to more produc-

tive duties (e.g., speed up computations of some critical search steps or initiate new search threads modelled on the most successful ones). A second possibility concerns performance statistics attached to individual solutions (or parts thereof) measuring, for example, the relative improvement of solutions generated by individual threads when starting from them. Another open question concerns the global search memories that could be inferred from the records built by individual searches and how the global search trajectory could then be inflected to yield superior results.

It has been noted that the pool of solutions constitutes a *population*. Then, other than the usual construction heuristics, population-based methods may be used to generate new solutions and improve the pool. Crainic and Gendreau (1999) report the development of such a hybrid search strategy combining their co-operative multi-thread parallel tabu search method with a genetic engine. The genetic algorithm initiates its population with the first elements from the central memory of the parallel tabu search. Asynchronous migration (migration rate = 1) subsequently transfers the best solution of the genetic pool to the parallel tabu search central memory, as well as solutions of the central memory towards the genetic population. Even though the strategy is not very elaborate, the hybrid appears to perform well, especially on larger problems. It is noteworthy that the genetic algorithm alone was not performing well and that it was the parallel tabu search procedure that identified the best results once the genetic method contributed to the quality of the central memory.

This combination of individual tabu search threads and population-based methods that run on the set of solutions in the central memory appears very promising and should be investigated in depth. (In more general terms, this strategy applies equally well to many meta-heuristic classes and is the object of research in the genetic and simulated annealing communities.) It also brings forth the issue of the parallelization of some advanced tabu search strategies, such as *path relinking* and *scatter search* (Glover and Laguna 1997).

In their sequential versions, both methods make use of a *reference* or *elite* set of solutions built gradually by a tabu search algorithm. At some moment during the search - that one may identify often as a search *diversification* point - a mechanism is called upon to create new solutions based on solutions in the reference set. These new solutions are then improved by the tabu search procedure, which also contributes to update the reference set. Of course, any of the strategies already presented in this paper may be applied to each individual search process. See, for example, the work of Gendreau *et al.* (1999) where a parallel adaptive memory approach is implemented while, simultaneously, the parallelism within each search thread is also exploited by decomposing the set of routes along the principles proposed in Taillard's work (1993). And, parallelism could be applied to speed up the generation of the new solutions. Yet, in our opinion, the most interesting avenue for the parallelization of path relinking

and scatter search approaches (and similar strategies) is to view them as mechanisms operating on the population of a central memory parallel co-operative method. These possibilities should be thoroughly investigated.

4 Perspectives and Research Directions

We have presented strategies to design parallel tabu search algorithms and surveyed developments and results in the area. Parallel tabu search methods offer the possibility to address problems more efficiently, both in terms of computing efficiency and solution quality. Low-level parallelization strategies appear often beneficial to speed up computation-intensive tasks, such as the evaluation of potential moves in the neighbourhood of a given solution. Moreover, such strategies may be advantageously incorporated into hierarchical parallel schemes where the higher level method either explores partitions of the solution domain or implements a co-operating multi-thread search.

Co-operation and multi-thread parallelization appear to offer the most interesting perspectives for tabu search, as for meta-heuristics in general. Asynchronous co-operative multi-thread search strategies constitute probably the strongest trend in parallel meta-heuristics, the results reported in the literature indicating that they offer better results than synchronous and independent searches. They also seem to offer the most interesting development avenue for advanced tabu search methods, including path relinking and scatter search. More theoretical and empirical work is still required in this field, however. Consequently, in the second part of the paper, we focussed on multi-search strategies. We discussed general design and implementation principles, pointed out a number of challenges and pitfalls, and identified trends and promising research directions.

Hybridization is another important trend in sequential and parallel meta-heuristics and recent studies tend to demonstrate that combining different meta-heuristics yields superior results. This opens up an exciting field of enquiry. What meta-heuristics to combine with tabu search? What role can each type of meta-heuristic play? What information is exchanged and how it is used in this context? How important a role path relinking and, specially, scatter search may play for parallel meta-heuristics in general? These are only a few of the questions that need to be explored.

It has been often noticed that co-operating parallel mechanisms bear little, if any, resemblance to the initial meta-heuristic one attempts to parallelize. This remark is true whether the individual search threads belong to the same class of meta-heuristics or not. A natural question then is whether co-operating multi-thread parallel methods

should form a "new", very broadly defined, class of meta-heuristics.

Is co-operative multi-thread parallelization a "new" meta-heuristic? Almost certainly, even though we face the challenge to properly define it. Is it a "new" tabu search class? Not as currently used. Indeed, the method misses global a search control mechanism and, especially, the memories that would guide this mechanism and are a central component of tabu search. The development of such memories and mechanisms is one of the most interesting challenges we face.

Acknowledgments

Funding for this project has been provided by the Natural Sciences and Engineering Council of Canada and by the Fonds F.C.A.R. of the Province of Québec.

References

[1] Aiex, R.M., Martins, S.L., Ribeiro, C.C., and Rodriguez, N.R. Asynchronous Parallel Strategies for Tabu Search Applied to the Partitioning of VLSI Circuits. Monografias em ciência da computação, Pontifícia Universidade Católica de Rio de Janeiro, 1996.

[2] Andreatta, A.A. and Ribeiro C.C. A Graph Partitioning Heuristic for the Parallel Pseudo-Exhaustive Logical Test of VLSI Combinational Circuits. *Annals of Operations Research*, 50:1–36, 1994.

[3] Badeau, P., Guertin, F., Gendreau, M., Potvin, J.-Y., and Taillard, É.D. A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows. *Transportation Research C: Emerging Technologies*, 5(2):109–122, 1997.

[4] Battiti, R. and Tecchiolli, G. Parallel Based Search for Combinatorial Optimization: Genetic Algorithms and TABU. *Microprocessors and Microsystems*, 16(7):351–367, 1992.

[5] Chakrapani, J. and Skorin-Kapov, J. A Connectionist Approach to the Quadratic Assignment Problem. *Computers & Operations Research*, 19(3/4):287–295, 1992.

[6] Chakrapani, J. and Skorin-Kapov, J. Massively Parallel Tabu Search for the Quadratic Assignment Problem. *Annals of Operations Research*, 41:327–341, 1993.

[7] Chakrapani, J. and Skorin-Kapov, J. Connection Machine Implementation of a Tabu Search Algorithm for the Traveling Salesman Problem. *Journal of Computing and Information Technology*, 1(1):29–36, 1993a.

[8] Chakrapani, J. and Skorin-Kapov, J. Mapping Tasks to Processors to Minimize Communication Time in a Multiprocessor System. In *The Impact of Emerging Technologies of Computer Science and Operations Research*, pages 45–64. Kluwer, Norwell, MA, 1995.

[9] Crainic, T.G. and Gendreau, M. Towards an Evolutionary Method - Cooperating Multi-Thread Parallel Tabu Search Hybrid. In S. Voß, S. Martello, C. Roucairol, and Osman, I.H., editors, *Meta-Heuristics 98: Theory & Applications*, pages 331–344. Kluwer, Norwell, MA, Montréal, QC, Canada, 1999.

[10] Crainic, T.G. and Gendreau, M. Cooperative Parallel Tabu Search for Capacitated Network Design. *Journal of Heuristics*, 2001. forthcoming.

[11] Crainic, T.G. and Toulouse, M. Parallel Strategies for Meta-heuristics. In F. Glover and G. Kochenberger, editors, *State-of-the-Art Handbook in Metaheuristics*. Kluwer, Norwell, MA, 2001. to appear; CRT-2001-06.

[12] Crainic, T.G., Toulouse, M., and Gendreau, M. Synchronous Tabu Search Parallelization Strategies for Multicommodity Location-Allocation with Balancing Requirements. *OR Spektrum*, 17(2/3):113–123, 1995a.

[13] Crainic, T.G., Toulouse, M., and Gendreau, M. Parallel Asynchronous Tabu Search for Multicommodity Location-Allocation with Balancing Requirements. *Annals of Operations Research*, 63:277–299, 1995b.

[14] Crainic, T.G., Toulouse, M., and Gendreau, M. Towards a Taxonomy of Parallel Tabu Search Algorithms. *INFORMS Journal on Computing*, 9(1):61–72, 1997.

[15] De Falco, I., Del Balio, R., and Tarantino, E. Solving the Mapping Problem by Parallel Tabu Search. Report, Istituto per la Ricerca sui Sistemi Informatici Paralleli-CNR, 1995.

[16] De Falco, I., Del Balio, R., Tarantino, E., and Vaccaro, R. Improving Search by Incorporating Evolution Principles in Parallel Tabu Search. In *Proc. Int. Conf. on Machine Learning*, pages 823–828, 1994.

[17] Fiechter, C.-N. A parallel tabu search algorithm for large travelling salesman problems. *Discrete Applied Mathematics*, 51(3):243–267, 1994.

[18] Garcia, B.L., J.-Y. Potvin, and Rousseau, J.M. A Parallel Implementation of the Tabu Search Heuristic for Vehicle Routing Problems with Time Window Constraints. *Computers & Operations Research*, 21(9):1025–1033, 1994.

[19] Gendreau, M., Guertin, F., Potvin, J.-Y., and Taillard, É.D. Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, 33(4):381–390, 1999.

[20] Glover, F. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research*, 1(3):533–549, 1986.

[21] Glover, F. Tabu Search – Part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[22] Glover, F. Tabu Search – Part II. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[23] Glover, F. Tabu Search and Adaptive Memory Programming – Advances, Applications and Challenges. In R.S. Barr, R.V. Helgason, and J. Kennington, editors, *Interfaces in Computer Science and Operations Research*, pages 1–75. Kluwer, Norwell, MA, 1996.

[24] Glover, F. and Laguna, M. Tabu search. In C.R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*, pages 70–150. Blackwell Scientific Publications, Oxford, 1993.

[25] Glover, F. and Laguna, M. *Tabu Search*. Kluwer, Norwell, MA, 1997.

[26] Glover, F., Taillard, É.D., and de Werra, D. A user's guide to tabu search. *Annals of Operations Research*, 41:3–28, 1993.

[27] Le Bouthillier, A. and Crainic, T.G. Parallel Co-operative Multi-thread Metaheuristic for the Vehicle Routing Problem with Time Window Constraints. Publication, Centre de recherche sur les transports, Université de Montréal, Montréal, QC, Canada, 2001.

[28] Malek, M., Guruswamy, M., Pandya, M., and Owens, H. Serial and Parallel Simulated Annealing and Tabu Search Algorithms for the Traveling Salesman Problem. *Annals of Operations Research*, 21:59–84, 1989.

[29] Martins, S.L., Ribeiro, C.C., and Rodriguez, N.R. Parallel Programming Tools for Distributed Memory Environments. Monografias em Ciência da Computação 01/96, Pontifícia Universidade Católica de Rio de Janeiro, 1996.

[30] Ouyang, M., Toulouse, M., Thulasiraman, K., Glover, F., and Deogun, J.S. Multi-Level Cooperative Search: Application to the Netlist/Hypergraph Partitioning Problem. In *Proceedings of International Symposium on Physical Design*, pages 192–198. ACM Press, 2000a.

[31] Ouyang, M., Toulouse, M., Thulasiraman, K., Glover, F., and Deogun, J.S. Multilevel Cooperative Search for the Circuit/Hypergraph Partitioning Problem. *IEEE Transactions on Computer-Aided Design*, 2000b. To appear.

[32] Porto, S.C.S., Kitajima, J.P.F.W., and Ribeiro, C.C. Performance Evaluation of a Parallel Tabu Search Task Scheduling Algorithm. *Parallel Computing*, 26:73–90, 2000.

[33] Porto, S.C.S. and Ribeiro, C.C. A Tabu Search Approach to Task Scheduling on Heteregenous Processors Under Precedence Constraints. *International Journal of High-Speed Computing*, 7:45–71, 1995.

[34] Porto, S.C.S. and Ribeiro, C.C. Parallel Tabu Search Message-Passing Synchronous Strategies for Task Scheduling Under Precedence Constraints. *Journal of Heuristics*, 1(2):207–223, 1996.

[35] Rego, C. and Roucairol, C. A Parallel Tabu Search Algorithm Using Ejection Chains for the VRP. In I.H. Osman and J.P. Kelly, editors, *Meta-Heuristics: Theory & Applications*, pages 253–295. Kluwer, Norwell, MA, 1996.

[36] Rochat, Y. and Taillard, É.D. Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of Heuristics*, 1(1):147–167, 1995.

[37] Schulze, J. and Fahle, T. A parallel algorithm for the vehicle routing problem with time window constraints. *Annals of Operations Reseach*, 86:585–607, 1999.

[38] Taillard, É.D. Robust Taboo Search for the Quadratic Assignment Problem. *Parallel Computing*, 17:443–455, 1991.

[39] Taillard, É.D. Parallel Iterative Search Methods for Vehicle Routing Problems. *Networks*, 23:661–673, 1993.

[40] Taillard, É.D. *Recherches itératives dirigées parallèles*. PhD thesis, École Polytechnique Fédérale de Lausanne, 1993a.

[41] Taillard, É.D. Parallel Taboo Search Techniques for the Job Shop Scheduling Problem. *ORSA Journal on Computing*, 6(2):108–117, 1994.

[42] Taillard, É.D., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J.-Y. A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. *Transportation Science*, 31:170–186, 1997.

[43] Toulouse, M., Crainic, T.G., and Gendreau, M. Communication Issues in Designing Cooperative Multi Thread Parallel Searches. In I.H. Osman and J.P. Kelly, editors, *Meta-Heuristics: Theory & Applications*, pages 501–522. Kluwer, Norwell, MA, 1996.

[44] Toulouse, M., Crainic, T.G., and Sansó, B. An Experimental Study of Systemic Behavior of Cooperative Search Algorithms. In S. Voß, S. Martello, C. Roucairol, and Osman, I.H., editors, *Meta-Heuristics 98: Theory & Applications*, pages 373–392. Kluwer, Norwell, MA, 1999.

[45] Toulouse, M., Crainic, T.G., Sansó, B., and Thulasiraman, K. Self-Organization in Cooperative Search Algorithms. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2379–2385. Omnipress, 1998.

[46] Toulouse, M., Crainic, T.G., and Thulasiraman, K. Global Optimization Properties of Parallel Cooperative Search Algorithms: A Simulation Study. *Parallel Computing*, 26(1):91–112, 2000.

[47] Toulouse, M., Glover, F., and Thulasiraman, K. A Multi-Scale Cooperative Search with an Application to Graph Partitioning. Report, School of Computer Science, University of Oklahoma, Norman, OK, 1998.

[48] Toulouse, M., Thulasiraman, K., and Glover, F. Multi-Level Cooperative Search. In P. Amestoy, P. Berger, M. Daydé, I. Duff, V. Frayssé, L. Giraud, and D. Ruiz, editors, *5th International Euro-Par Parallel Processing Conference*, volume 1685 of *Lecture Notes in Computer Science*, pages 533–542. Springer-Verlag, Berlin, 1999.

[49] Voß, S. Tabu Search: Applications and Prospects. In D.-Z. Du and P.M. Pardalos, editors, *Network Optimization Problems*, pages 333–353. World Scientific Publishing Co., Singapore, 1993.